(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 April 2001 (26.04.2001)

PCT

(10) International Publication Number
WO 01/29726 A2

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US00/25191

(22) International Filing Date:
15 September 2000 (15.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/159,547	15 October 1999 (15.10.1999)	US
09/473,052	28 December 1999 (28.12.1999)	US

(71) Applicant: NET PERCEPTIONS, INC. [US/US]; 7901 Flying Cloud Drive, Eden Prairie, MN 55344-7905 (US).

(72) Inventors: LOUCKS, Russell, J.; 4767 London Lane, Eagan, MN 55122 (US). NORTON, Jeremy, J.; 475 Owasso Hills Drive, Roseville, MN 55113 (US). SALMONSON, Steven, P.; 2816 Nevada Avenue S., Saint Louis Park, MN 55426 (US). SCHUTTA, Nancy, K.; 1221 Brighton Square, New Brighton, MN 55112 (US). THORN, Eric; 6254 Harbrough Court, Eden Prairie, MN 55346 (US).

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

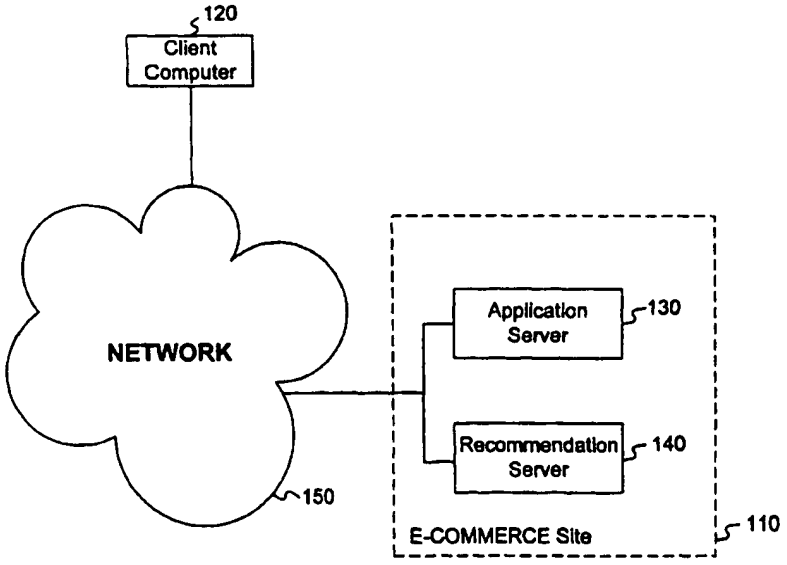
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SHOPPING SESSION APPLICATION FRAMEWORK



(57) Abstract: Methods and systems consistent with the present invention solve the inherent problems with current recommender systems by providing a shopping session application framework facility to access recommendation engines from e-commerce sites. Specifically, the framework application easily adds recommendation engine capabilities to existing e-commerce applications by providing a set of interfaces from the e-commerce application to the recommendation engine. The interfaces may record user activity during online shopping sessions and provide access to a recommendation engine for various predictions.

WO 01/29726 A2

SHOPPING SESSION APPLICATION FRAMEWORK

RELATED APPLICATIONS

Provisional U.S. Patent Application No. 60/159,547, entitled "Shopping Session Application Framework," filed October 15, 1999 is relied upon and is incorporated by reference in this application.

BACKGROUND OF THE INVENTION

A. Field of the Invention

This invention relates generally to data processing systems, and more particularly, to recommendation systems.

B. Description of the Related Art

Recommender systems predict the preferences of users based on attributes known about the user or a past history of preferences or consumptions by the user. For example, a recommender system may predict that a user will like the movie "Titanic" because the user previously indicated a preference for such other epic movies as "Lawrence of Arabia" or "Ben Hur."

Because of their ability to predict user preference, recommender systems are becoming widely used in e-commerce business activities. For example, systems that make personalized recommendations are used as a marketing tool to turn "window shoppers" into buyers, increase cross-sells and up-sells, and deepen customer loyalty. Recommender systems allow e-commerce administrators to take advantage of customer databases to provide valuable personalized service to customers.

One type of e-commerce application that employs recommender systems are shopping cart applications. A shopping cart is the interface between a company's web site and its deeper infrastructure, allowing consumers to select merchandise; review what they have selected; make necessary modifications or additions; and purchase the merchandise. Shopping carts can be sold as independent pieces of software so companies can integrate them into their own unique online solution, or they can be offered as a feature from a commercial service that will create and host a company's e-commerce site.

Although e-commerce sites use recommender systems in various e-commerce applications, recommender systems are complicated and require special adaptations for

each e-commerce application. That is, recommender systems contain recommendation engines that require front-end interfaces to access. Moreover, since recommendation engines are inherently different, each time an e-commerce application requires a recommendation engine, the e-commerce administrator must create new software to access the interfaces of the recommendation engine.

Therefore, there exists a problem with existing recommender systems that, although able to provide robust recommendations to e-commerce application, they are cumbersome and there is no standard way to access them. Consequently, recommender systems are not used to their full potential. It is therefore desirable to improve upon existing recommender systems.

SUMMARY OF THE INVENTION

Methods and systems consistent with the present invention solve the inherent problems with current recommender systems by providing a shopping session application framework facility to access recommendation engines from e-commerce sites. Specifically, the framework application easily adds recommendation engine capabilities to existing e-commerce applications by providing a set of interfaces from the e-commerce application to the recommendation engine. The interfaces record user activity during online shopping sessions and provide access to a recommendation engine for various predictions.

Consistent with the principles of the present invention, a method for providing access from an e-commerce application to a recommendation engine with an interface is disclosed. The method maintains a session object including data corresponding to user activity. Once the session object is created, the method requests a recommendation from the interface based on the user activity, and updates the recommendation engine with the contents of the session object through the interface. Consistent with the principles of the present invention, a method for personalizing a shopping session in an e-commerce application is provided. The method creates a shopping session object, maintains a log of a user's activity, adds the activity to the shopping session object, and transmits a request for a prediction to the recommendation engine based on the contents of the shopping session object. In response to the request, the method receives prediction results from the recommendation engine in the e-commerce application, and updates a

database to include the user's activity.

Consistent with the principles of the present invention, an e-commerce application server that provides access to a recommendation engine is provided. The application server contains a session object that contains user activity on the e-commerce application server, means for recording the user activity in the session object, means for requesting a recommendation from a recommendation engine based on the contents of the session object, and means for updating the recommendation engine with the contents of the session object.

Consistent with the principles of the present invention, a method for providing access from an e-commerce application to a recommendation engine using a framework application is provided. The method receives a request to access a recommendation engine from the e-commerce application, maintains a session object at the framework application including data corresponding to user activity, and requests a recommendation from the framework application to the recommendation engine based on the received request and the user activity. In response to the request, the method receives a response from the recommendation engine at the framework application that includes results, and forwards the results from the framework application to the e-commerce application.

Consistent with principles with the present invention, a data processing system is provided. The data processing system contains a memory containing a framework application program and a processor to run the program. The program receives a request to access a recommendation engine, maintains a session object at the framework application including data corresponding to user activity, requests a recommendation from the recommendation engine based on the received request and the user activity, receives a response from the recommendation engine that includes results, and forwards the results to an e-commerce application.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an implementation of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

Figure 1 depicts a data processing system suitable for practicing methods and systems consistent with the present invention;

Figure 2 depicts a more detailed diagram of the client computer depicted in Fig. 1;

Figure 3A depicts a more detailed diagram of the recommendation server depicted in Fig. 1;

Figure 3B depicts a more detailed diagram of the e-commerce server depicted in Fig. 1;

Figure 4 depicts a layering connection between the e-commerce application and the recommendation engine.

Figure 5 depicts a flow chart of the steps performed by the data processing system of Fig. 1 when providing access to a recommendation engine; and

Figure 6 depicts a shopping session object consistent with methods and systems of the present invention.

DETAILED DESCRIPTION

The following detailed description of the invention refers to the accompanying drawings. Although the description includes exemplary implementations, other implementations are possible, and changes may be made to the implementations described without departing from the spirit and scope of the invention. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Overview

Methods and systems consistent with the present invention solve the inherent problems with current recommender systems by providing a shopping session application framework facility to access recommendation engines from e-commerce sites. Specifically, the framework application easily adds recommendation engine capabilities to existing e-commerce applications by providing a set of interfaces from the e-commerce application to the recommendation engine. The interfaces provide a connection to record user activity during online shopping.

System Components

Fig. 1 depicts a data processing system 100 suitable for practicing methods and systems consistent with the present invention. Data processing system 100 comprises a client computer 120 connected to an application server 130 and a recommendation server 140 via a network 150, such as the Internet. A user uses client computer 120 to provide various information to application server 130. Together, application server 130 and recommendation server 140 form an e-commerce site 110. Recommendation server 140 interfaces with application server 130 using an Application Program Interface (API). An API is a set of routines, protocols, or tools for communicating with software applications, in this case a recommendation engine operating on server 140. The APIs provide efficient access to recommendation server 140 without the need for an e-commerce operators to create software that interfaces with the recommendation engine.

Although only one client computer 120 is depicted, one skilled in the art will appreciate that data processing system 100 may contain many more client computers and additional client sites. One skilled in the art will also appreciate that application server 140 may be located at various places on network 150, including client computer 120.

Figure 2 depicts a more detailed diagram of client computer 120, which contains a memory 220, a secondary storage device 230, a central processing unit (CPU) 240, an input device 250, and a video display 260. Memory 220 includes browser 222 that allows users to interact with application server 130 by transmitting and receiving files, such as web pages. A web page may include images or instructions to obtain recommendation requests from a user using hypertext markup language (HTML), Java, or other techniques. An example of a browser suitable for use with methods and systems consistent with the present invention is the Netscape Navigator browser, from Netscape.

As shown in Figure 3A, recommendation server 140 includes a memory 310, a secondary storage device 320, a CPU 330, an input device 340, and a video display 350. Memory 310 includes recommendation engine 312, which determines if an item should be recommended to a user. Recommendation engine 312 may use many different techniques to generating recommendations from user interest profiles. One such example that is used to generate recommendations are automated collaborative filtering described in Resnick, Iacovo, Susha, Bergstrom, and Riedl, "GroupLens: An Open Architecture For Collaborative Filtering Of Netnews," Proceedings of the 1994

Computer Supported Collaborative Work Conference (1994). Other recommendation techniques are described in U.S. application serial no. 08/729,787, filed October 8, 1996, U.S. application serial no. 08/733,806, filed October 18, 1996, attorney docket no. 7744-6000, filed September 23, 1999, and attorney docket no. 7744-0009, filed September 24, 1999, all incorporated by reference. Recommender systems may also be based on well-known CF systems, logical rules derived from data, or on statistical or machine learning technology. For example, a recommender system may use well-known rule-induction learning, such as Cohen's Ripper, to learn a set of rules from a collection of data as described in Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., "Combining Collaborative Filtering with Personal Agents for Better Recommendations," Prceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99). Recommender systems may also be based on well-known data mining techniques that include a variety of supervised and unsupervised learning strategies and produce "surprising" results expressed as associations or rules embedded in a data set, such as results obtained from web page logs, or purchase histories stored in e-commerce databases. Recommender systems may also contain rating functions programmed by a system administrator. The rating functions are either a formula or a table of ratings that determines business goals (e.g., the formula may specify a low rating for low-stock and out-of-stock items). These mentioned systems also require user data as input to produce personalized recommendations for users.

Recommendation engine 312 also permits access from application server 130 for transmitting and receiving recommendation requests and populating a recommendation database. To provide this access, recommendation engine 312 includes an API 314. API 314 contains APIs that enable application server 130 to access parts of recommendation engine 312. Secondary storage device 320 includes a database 322 that stores user activity on e-commerce site 110.

As shown in Figure 3B, application server 130 includes a memory 360, a secondary storage device 370, a CPU 380, an input device 390, and a video display 395. Memory 360 includes an e-commerce application 362 that provides a web application with a shopping cart to the user. For example, e-commerce application 362 may be an online bookstore or an online grocer. E-commerce application 362 also contains well-known web server software (not shown) to transmit and receive files to the user. Also

in memory 360 is framework application 364 that stores various user activities. For example, framework application 364 may store a users "click-throughs," purchases, quantities, totals, and information obtained from various web page logs. Framework application 364 also provides an interface for e-commerce application 362 to access recommendation engine 312. By providing this interface, recommendations can be easily integrated into an e-commerce application without having the e-commerce administrator create a customized front-end to recommendation engine 312.

Alternatively, framework application 364 may be contained on a framework application server (not shown). E-commerce application 362 may access software on the framework application server via the APIs provided by the framework application. The framework application server allows various well-known e-commerce applications to access the framework application with no modification to the e-commerce application. As shown in Fig. 4, framework application 364 is a layer between e-commerce application 362 and recommendation engine 312. For example, an online bookstore could include recommendations obtained from recommendation engine 312 via the framework application server. The e-commerce application could communicate with the framework application server via APIs provided by the framework application. When API queries are received at a framework application server, the server communicates directly with the recommendation engine.

Secondary storage device 370 includes a database 372 that stores users' shopping carts as shopping session objects 374, further described below, and the details of any placed orders in an order file 376. For example, order file 376 may store user identification, items purchased, quantity purchased, and price paid. One skilled in the art will appreciate that order file 376 may contain other details. Also in secondary storage device 370 is a well-known Online Analytical Processing (OLAP) database 378. The OLAP database provide a convenient way to access summarized data. For example, if ten users ordered 100 disks each, OLAP database 378 may store the total number of disks (1000) ordered.

Although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects may be stored on or read from other computer-readable media, such as secondary storage devices, like hard disks, floppy disks, and CD-ROM; a carrier wave received from a network like the Internet; or

other forms of ROM or RAM. Additionally, although specific components and programs of client computer 120, recommendation server 140, and application server 130 have been described, one skilled in the art will appreciate that these may contain additional or different components or programs.

Access Process

Figure 5 depicts a flow chart of the steps performed when accessing a recommendation server 140. The access process is initiated, for example, by a user accessing application 362 on application server 130 (step 502). A user may access application 362 to browse for a book, or purchase items from the online grocer. Once accessed, application 362 queries framework application 364 to create a new shopping session object 374 for the user (step 504). Figure 6 shows a shopping session object 374 consistent with methods and systems of the present invention. Object 374 contains an object reference number, a unique user ID, a list of all web pages viewed by the user while accessing application 362 called a "click-list," and a list of all items placed in the user's shopping cart.

To create object 374, framework application 364 assigns a unique reference number to a new object 374 (step 506), and provides the reference number to application 362 (step 508). Application 362 uses the reference number to reference the object in an API call to framework application 364 each time information is added or removed from the object.

Once object 374 is created and a reference number is obtained, application 362 monitors user activity (step 510). To do so, each time a user views a web page or selects an item to place in the shopping cart, e-commerce application 362 queries framework application 364 with an API to update object 374 (step 512). For example, if a user clicks on a particular web page, application 362 may query framework application 364 with a "record_click" API. The record_click API is a request for framework application 364 to add the viewed web page to the click list in object 374. If a user places an item in the shopping cart, application 362 may query with an "add_to_basket" API. The add_to_basket API is a request for framework application 364 to add the selected item to the shopping cart list in object 374. Framework application 364 records the user activity in object 374 to provide a recommendation based on the user activity when the

user nears completion of a shopping session and "checks-out" of e-commerce application 362. One skilled in the art will appreciate that framework application 364 may record other user activity, such as monitoring objects relating to an e-commerce promotion.

Once the user finishes accessing application 362, the application queries framework application 364 with a "predict" API (step 516). The predict API returns a list of items to application 362 that may interest the user. The predict API obtains information from object 374 and communicates directly with recommendation engine 312 through API interface 314. The predict API queries recommendation engine 312 to return a recommendation based on the user activity stored in shopping cart session object 374. A recommendation may be based on the items in the shopping cart, click-throughs, historical purchases for the user, cross sell lists, or from a set of shopping session objects. For example, to receive a recommendation from recommendation engine 312 using click-throughs for object number "123," application 362 may use the predict API as follows: call predict_API(click-through, session 123). When the predict API call is received at framework application 364, framework application transmits a second API request to API interface 314 that includes data from object "123" and the term "click-through."

Once the recommendation request is received at recommendation engine 312, the engine may use the different techniques described above to generate a list of recommendations (step 518).

In response to the API call from framework application 364, recommendation engine 312 returns a list of items to framework application 264 which submits the results in a format readable to application 362. Applicant 362 may display the list of items to the user before proceeding to check-out (step 520). Alternatively, if enough items are not located in recommendation database 322, a default list may be used. For example, if object 364 does not contain enough user activity for recommendation engine 312 to provide a worthwhile recommendation, application 362 may obtain a default recommendation from framework application 364 by calling the predict API as follows: call predict_API(default, session 123).

Once the recommended items are displayed to the user and the user has proceeded to check-out, application 362 may query framework application 364 with a "buy" API that updates various information in recommendation server 140 and object

374 (step 522). When framework application 364 receives the buy API call, the framework clears shopping session object 374 and initiates a call to recommendation engine 312 to update recommendation database 322. Even if the user does not purchase an item, recommendation engine 312 adds the user activity to database 322. In doing so, database 322 may provide more accurate recommendations to future users.

Also, once the user has proceeded to check out, application 362 may update order file 376 and OLAP database 378. If the user purchases an item, a new record is appended to order file 376 to indicate that an order is pending. Also in application server 130, the total order is added to OLAP database 378.

Conclusion

Methods and systems consistent with the present invention solve the inherent problems with current recommender systems by providing a shopping session application framework facility for e-commerce sites. Specifically, the framework may easily add recommendation engine capability to existing shopping cart applications. To do so, the framework monitors user activity during online shopping sessions and provides access to a recommendation engine for various predictions.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone.

WHAT IS CLAIMED IS:

1. A method for providing access from an e-commerce application to a recommendation engine through an interface, executed in a data processing system, comprising the steps of:

maintaining a session object including data corresponding to user activity;

requesting a recommendation from the interface based on the user activity, wherein the interface communicates with the recommendation engine; and

updating the recommendation engine with the contents of the session object through the interface.

2. The method of claim 1, wherein the interface connects to a framework application server.

3. The method of claim 1, wherein maintaining a session object further includes:

requesting an object reference to the session object from the interface; and

passing the object reference to the e-commerce application.

4. The method of claim 1, wherein maintaining a session object further includes:

recording user click-throughs and user purchases in the e-commerce application.

5. The method of claim 1, wherein maintaining a session object further includes:

storing, in the session object, the user activity.

6. The method of claim 1, wherein requesting a recommendation further includes:

querying the interface from the e-commerce application with an API; and

receiving a recommendation list from the recommendation engine.

7. The method of claim 6, wherein the API includes a function based on the user click-throughs.

8. The method of claim 6, wherein the API includes a function based on user purchases.

9. The method of claim 6, wherein the API includes a function based on a user's identity.

10. The method of claim 6, wherein the API includes a function based on a default list.

11. The method of claim 1, wherein updating the recommendation engine further includes:

adding the contents of the session object to a recommendation engine database

using an API; and

adding the user purchases to an e-commerce application database.

12. The method of claim 1, wherein the session object contains purchased items, and wherein the method further includes:

adding totals of purchased items in the shopping session object to an OLAP database; and

adding information corresponding to the purchased item to an order database.

13. A method for personalizing a shopping session in an e-commerce application, comprising the steps of:

creating a shopping session record by accessing a recommendation engine;

maintaining a log of a user's activity and adding information reflecting the activity to the shopping session record;

transmitting a request for a prediction to the recommendation engine based on the information in the shopping session record;

receiving prediction results from the recommendation engine in the e-commerce application; and

updating a database to include the user's activity.

14. The method of claim 13, wherein creating a shopping session object further includes:

requesting an object reference to the shopping session object from the recommendation engine; and

passing the object reference to the e-commerce application.

15. The method of claim 13, wherein transmitting a request for a prediction further includes:

including in the request a prediction method that uses the user's activity.

16. The method of claim 13, wherein the shopping session object contains purchased items, and wherein updating a database further includes:

adding totals of purchased items in the shopping session object to an OLAP database; and

adding information corresponding to the purchased item to an order database.

17. The method of claim 13, wherein updating a database further includes:

adding the user's activity to a recommendation engine database.

18. An e-commerce application server for providing access to a recommendation engine, comprising:

a session object that contains user activity on the e-commerce application server;

means for recording the user activity in the session object;

means for requesting a recommendation from a recommendation engine based on the contents of the session object; and

means for updating the recommendation engine with the contents of the session object.

19. The system of claim 18, wherein means for requesting a recommendation further includes:

means for accessing an interface to the recommendation engine from the e-commerce application using an API; and

means for receiving a recommendation list from the interface.

20. The system of claim 19, wherein the interface connects to a framework application server.

21. The system of claim 19, wherein the API includes a function based on the user click-throughs.

22. The system of claim 19, wherein the API includes a function based on user purchases.

23. The system of claim 19, wherein the API includes a function based on a user's identity.

24. The system of claim 19, wherein the API includes a function to create a default list.

25. The system of claim 18, further comprising:

a framework application server to facilitate communicating between the e-commerce application server and the recommendation engine.

26. The system of claim 18, further comprising:

means for requesting an object reference to the session object from a framework application; and

means for passing the object reference to the e-commerce application.

27. The system of claim 18, wherein means for recording user activity further includes:

means for recording user click-throughs and user purchases in the e-commerce application.

28. The system of claim 18, wherein means for updating the recommendation engine further comprises:

means for adding the contents of the session object to a recommendation engine database with an API using an interface; and

means for adding the user purchases to an e-commerce application database.

29. The system of claim 18, wherein the session object contains purchased items, and wherein the system further comprises:

means for adding totals of purchased items in the shopping session object to an OLAP database; and

means for adding information corresponding to the purchased item to an order database.

30. A system for personalizing a shopping session in an e-commerce application, comprising:

a shopping session that is created by accessing an interface to a recommendation engine;

means for maintaining a log of a user's activity and adding the activity to the shopping session object;

means for transmitting a request for a prediction to the interface based on the contents of the shopping session object;

means for receiving prediction results from the interface in the e-commerce application; and

a database associated with the e-commerce application that includes the user's activity.

31. The system of claim 30, wherein the interface is connected to a framework application server.

32. The system of claim 30, further comprising:

means for requesting an object reference to the shopping session object from the interface; and

means for passing the object reference to the e-commerce application.

33. The system of claim 30, wherein means for transmitting a request for a prediction

further comprises:

means for including in the request a prediction method to apply to the user's activity.

34. The system of claim 30, wherein the shopping session object contains purchased items, and wherein the system further comprises:

means for updating the database by adding totals of purchased items in the shopping session object to an OLAP database and adding information corresponding to the purchased item to an order database.

35. The system of claim 34, wherein updating a database further includes:

means for adding the user's activity to a recommendation engine database.

36. A method for providing access from an e-commerce application to a recommendation engine using a framework application, executed in a data processing system, comprising the steps of:

receiving a request to access the recommendation engine from the e-commerce application at the framework application;

maintaining a session object at the framework application including data corresponding to user activity;

requesting a recommendation from the framework application to the recommendation engine based on the received request and the user activity;

receiving a response from the recommendation engine at the framework application that includes results; and

forwarding the results from the framework application to the e-commerce application.

37. A data processing system comprising:

a memory containing a framework application program that receives a request to access a recommendation engine, that maintains a session object at the framework application including data corresponding to user activity, that requests a recommendation from the recommendation engine based on the received request and the user activity, that receives a response from the recommendation engine that includes results, and that forwards the results to an e-commerce application;

a processor configured to run the framework application program.

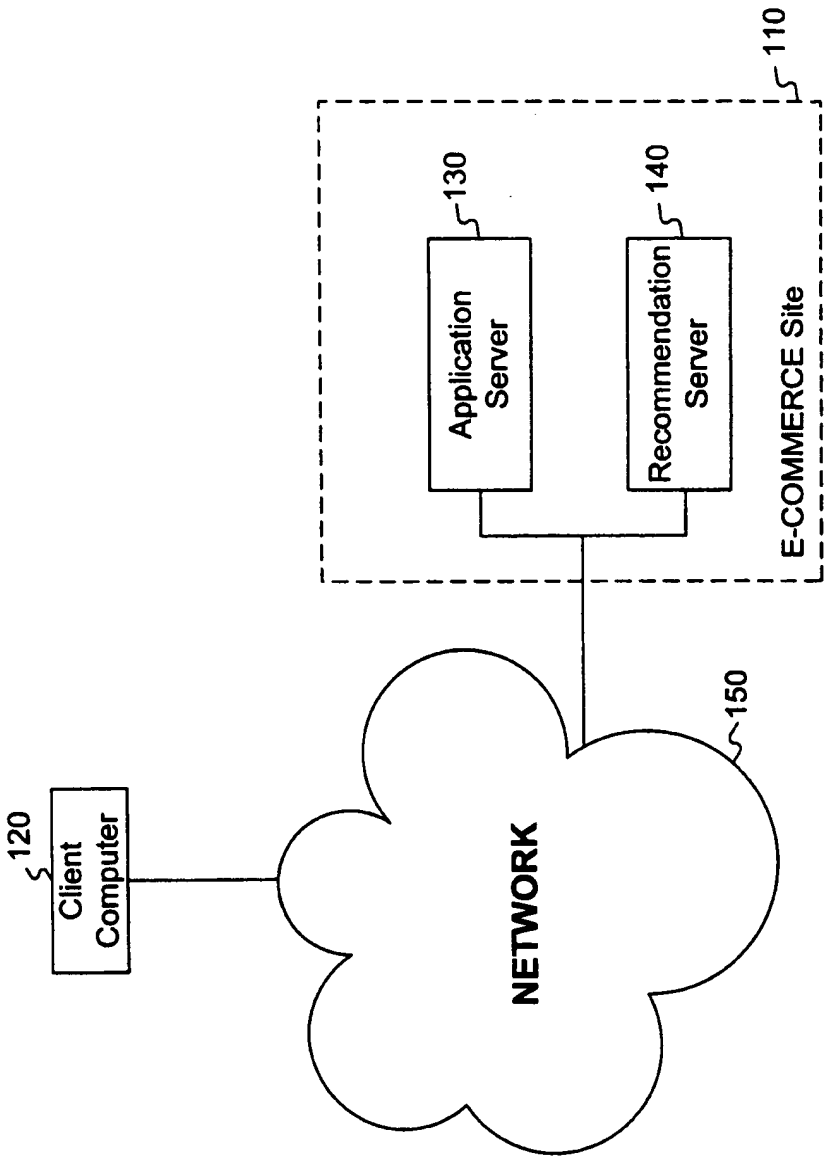

100
110
E-COMMERCE Site
130
Application Server
140
Recommendation Server
150
NETWORK
120
Client Computer

FIG. 1

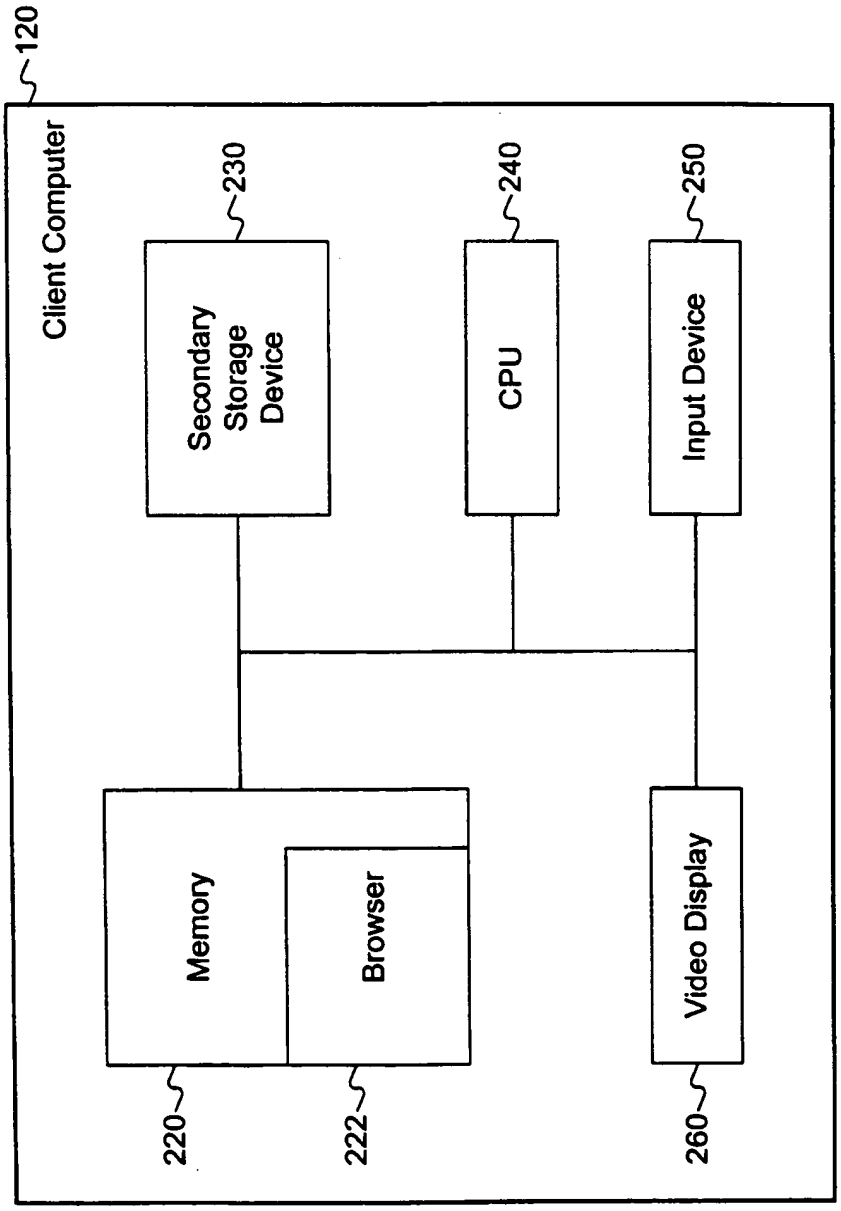

120
Client Computer
230
Secondary Storage Device
240
CPU
250
Input Device
Memory
Browser
Video Display
220
222
260

FIG. 2

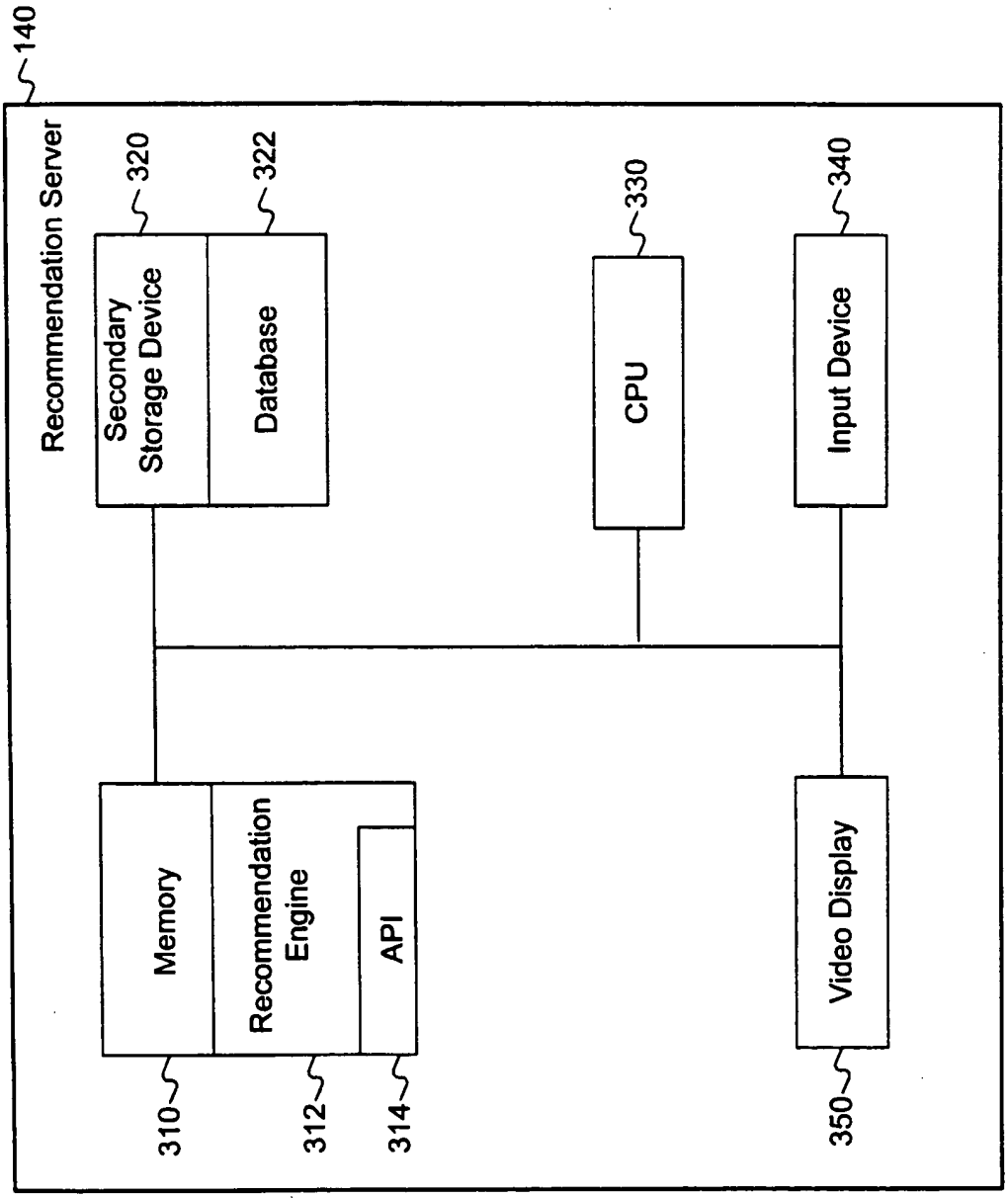

140
Recommendation Server
320
Secondary Storage Device
322
Database
330
CPU
340
Input Device
Memory
Recommendation Engine
API
Video Display
310
312
314
350

FIG. 3A

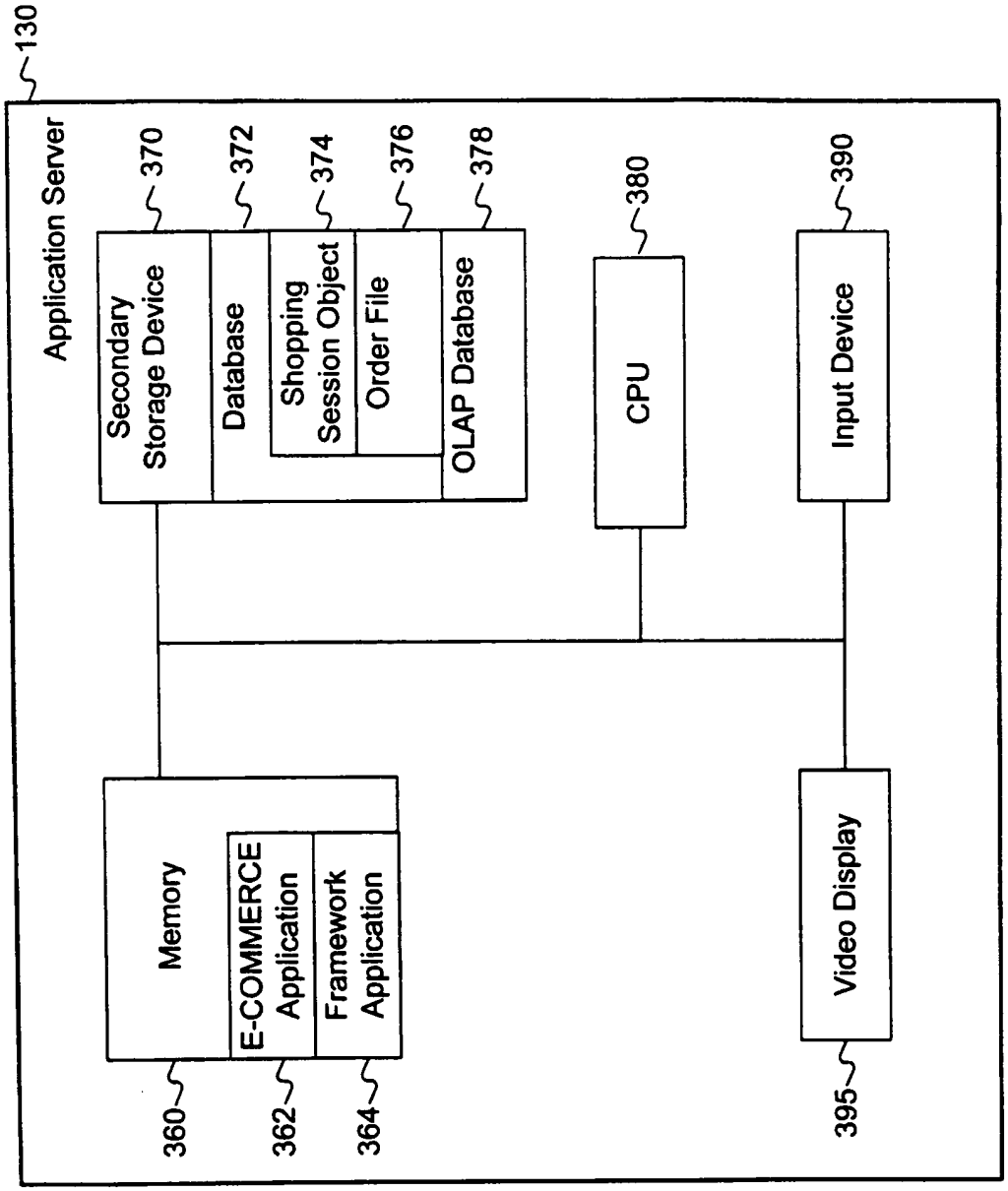

130
Application Server
Secondary Storage Device
370
Database
372
Shopping Session Object
374
Order File
376
OLAP Database
378
CPU
380
Input Device
390
Memory
E-COMMERCE Application
Framework Application
360
362
364
Video Display
395

FIG. 3B

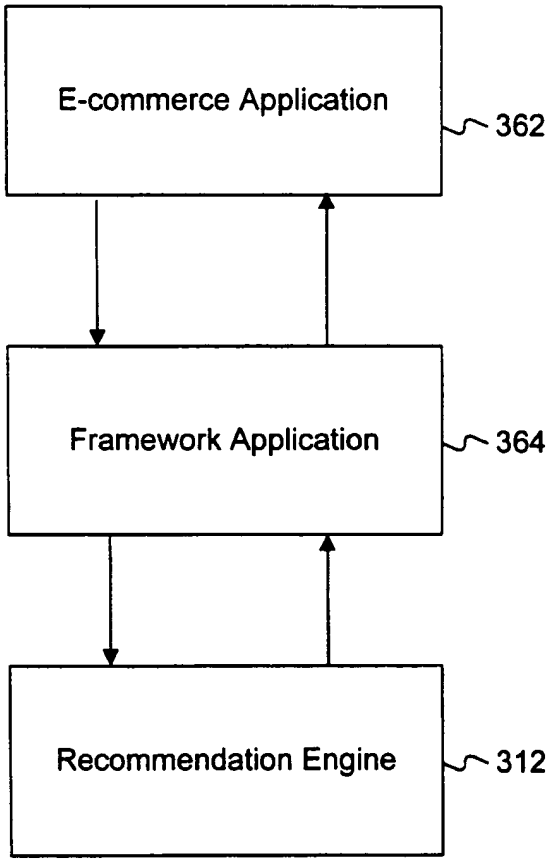

E-commerce Application
362
Framework Application
364
Recommendation Engine
312

Fig. 4

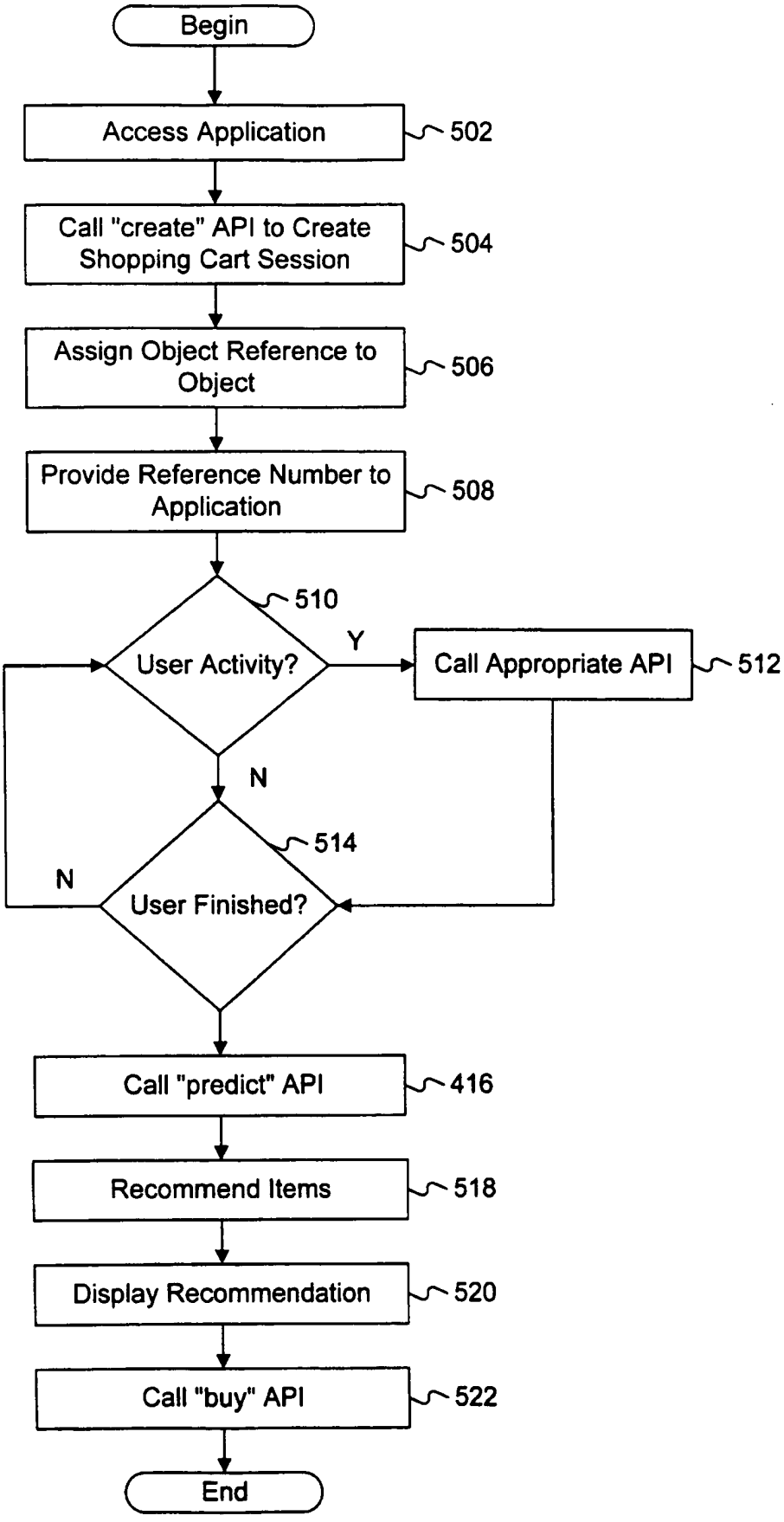

Begin
Access Application
502
Call "create" API to Create Shopping Cart Session
504
Assign Object Reference to Object
506
Provide Reference Number to Application
508
510
User Activity?
Y
Call Appropriate API
512
N
514
N
User Finished?
Call "predict" API
416
Recommend Items
518
Display Recommendation
520
Call "buy" API
522
End

Fig. 5

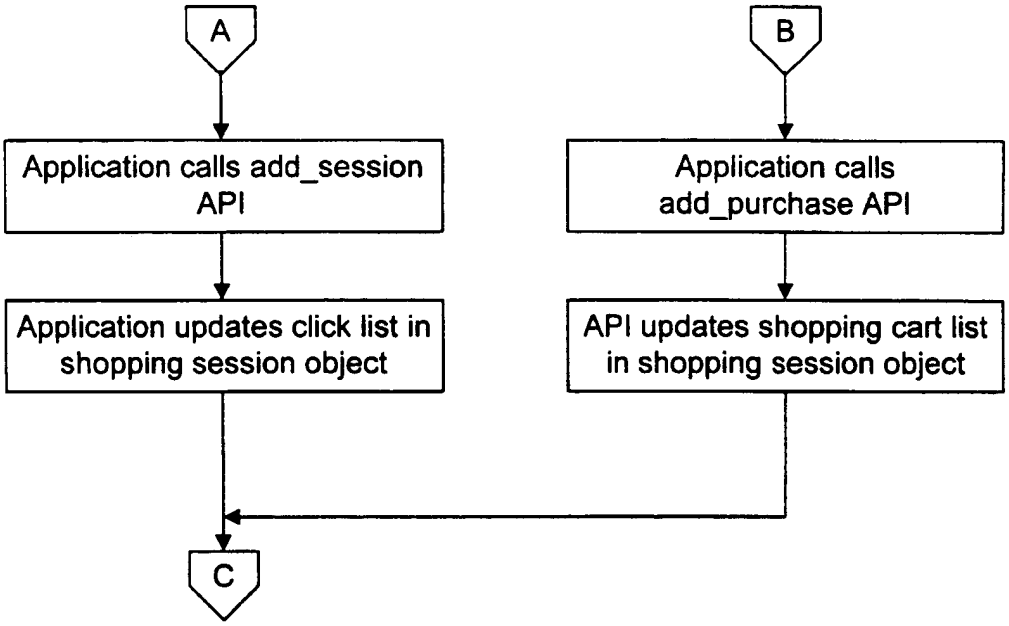

A
B
Application calls add_session API
Application calls add_purchase API
Application updates click list in shopping session object
API updates shopping cart list in shopping session object
C

Fig. 5B

374

OBJECT NUMBER	USER ID	CLICK LIST	SHOPPING CART LIST
123	01234	Sport.html Football.html Soccer.html	Football, #1234 Helmet, #2934

FIG. 6